

設備認識 與Modbus基礎實務

LIVING 3.0 智慧化居住空間展示中心
108 年智慧生活應用工作坊



課程目標

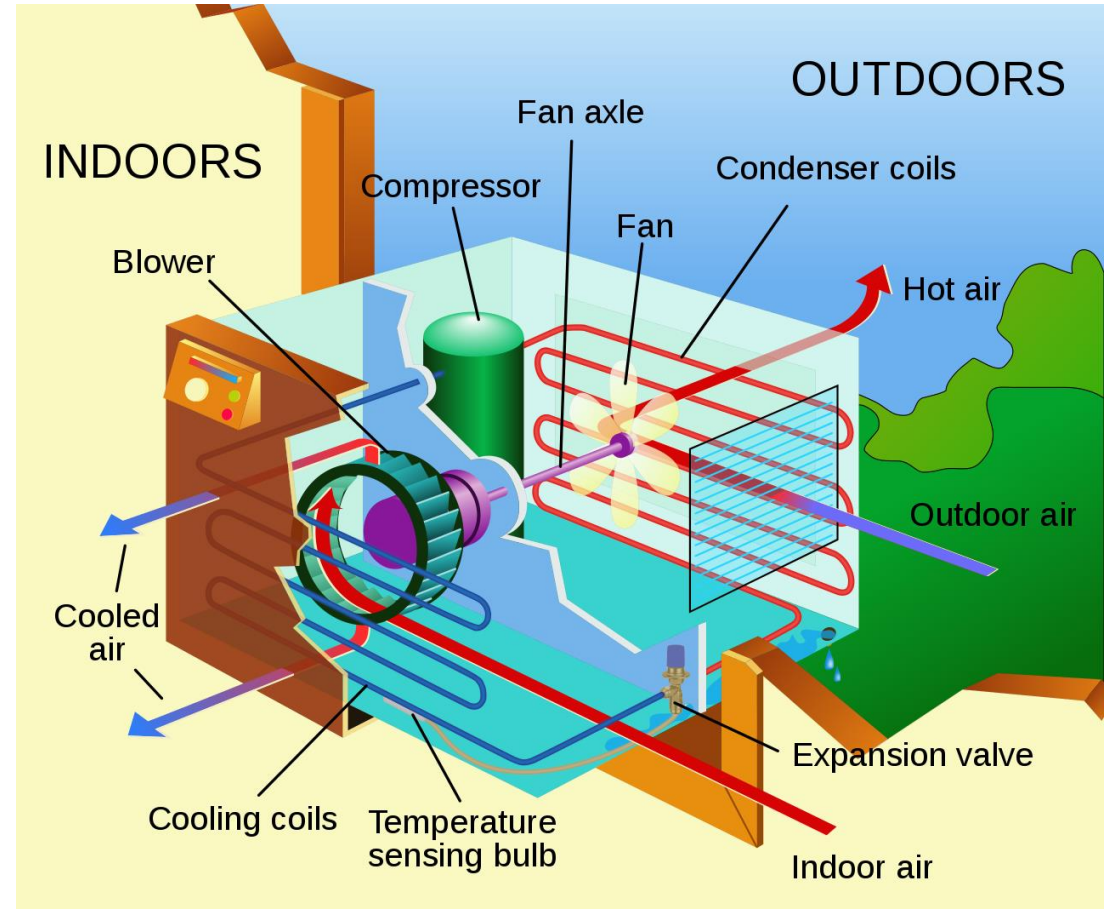
- 認識居家空調設備。
- 理解系統整合基本概念。
- 理解 Modbus 通訊協定。
- 利用 PyModbus 實踐基本 Modbus 通訊。

事前準備

- SSH 伺服連線工具
 - MobaXterm。
 - Mac 或 Linux 內建之 ssh 殼層應用程式。
- Python 套件
 - > pip install pymodbus
- 非必要工具
 - Wireshark 網路封包監聽用。

居家空調設備

- 空氣調節
 - 控制空氣之物理特性滿足人員舒適或其他需求。
- 常見居家空調設備
 - 冷（暖）氣
 - 提供製冷製熱
 - 可控制溫度
 - 全熱交換器
 - 提供新鮮外氣
 - 可控制汙染物（如 CO₂）濃度



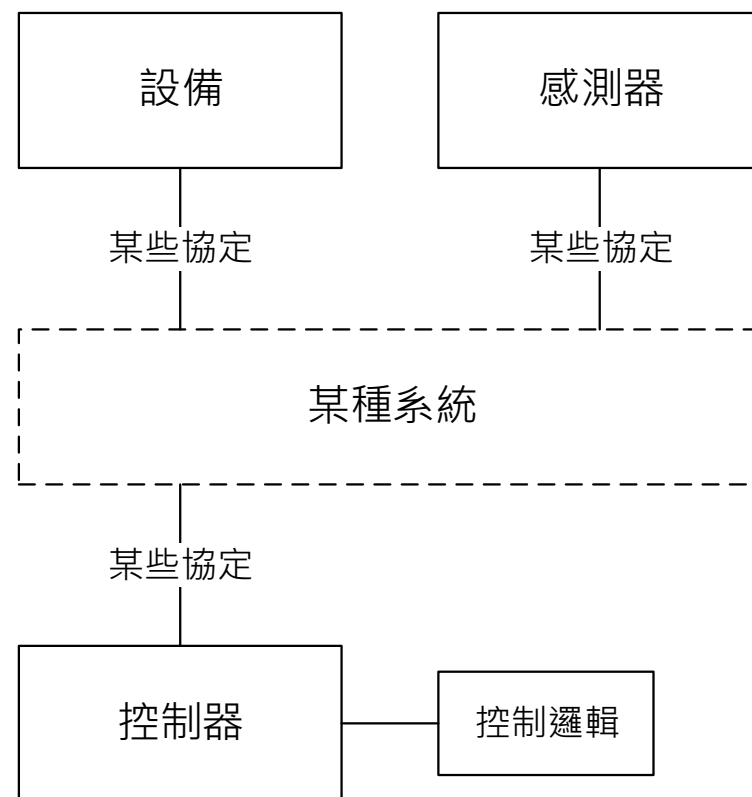
圖源：[Wikipedia](https://en.wikipedia.org/wiki/Refrigeration_cycle#/media/File:Refrigeration_cycle.svg)

控制系統設計通論

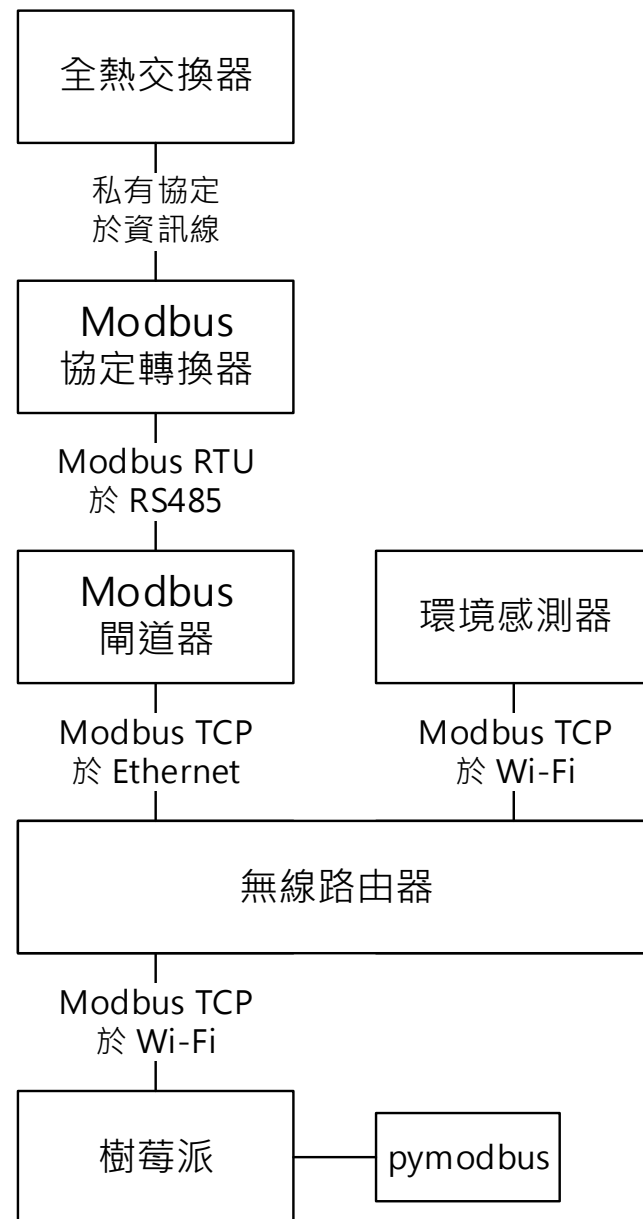
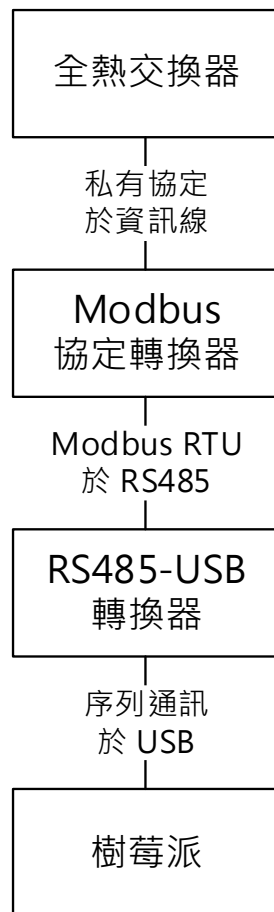
- 流程：

1. 發掘問題。
 - 我想增進室內空氣品質。
 - 我想採用新風引入的方式進行。
2. 盤點資源（硬體與技術）。
 - 我有個全熱交換機。
 - 說明書表示其採用 Modbus 通訊。
 - 據說 Modbus 通訊等等老師會教。
3. 設計系統。
4. 實踐系統。

控制型系統基本框架



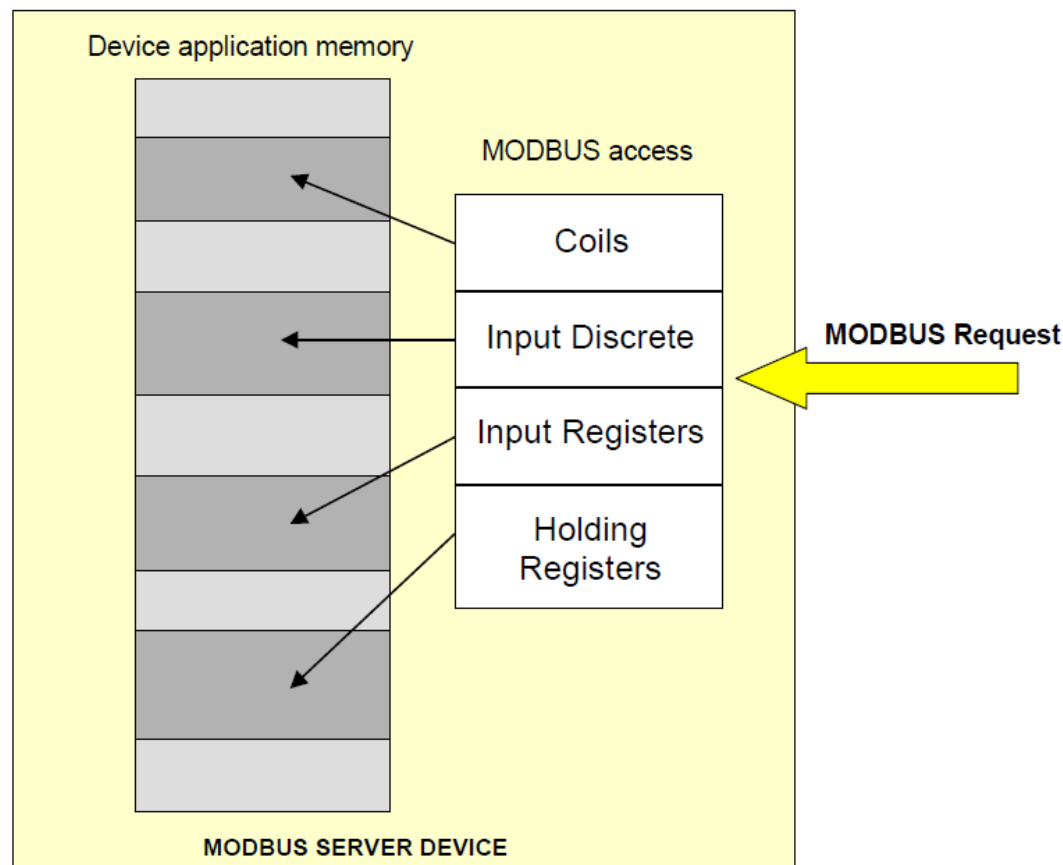
課程整合架構



Modbus 協定

- 為整合 PLC（可程式化控制器）通訊設計之協定，是現今工業控制領域的主流標準。
- 該協定包含設備記憶體配置與傳訊方法之規範。
- 不同的網路層有不同的擴展：
 - Modbus RTU（通常乘載於 RS-485 實體層）。
 - Modbus TCP（乘載於 Ethernet 上並以 TCP/IP 方式封裝）。

Modbus 記憶體配置



- 四種資料類型抽象

- 線圈

- 數位輸出點 (Digital Output)
 - 每位址1 bit 寬

- 離散輸入

- 數位輸出點 (Digital Input)
 - 每位址1 bit 寬

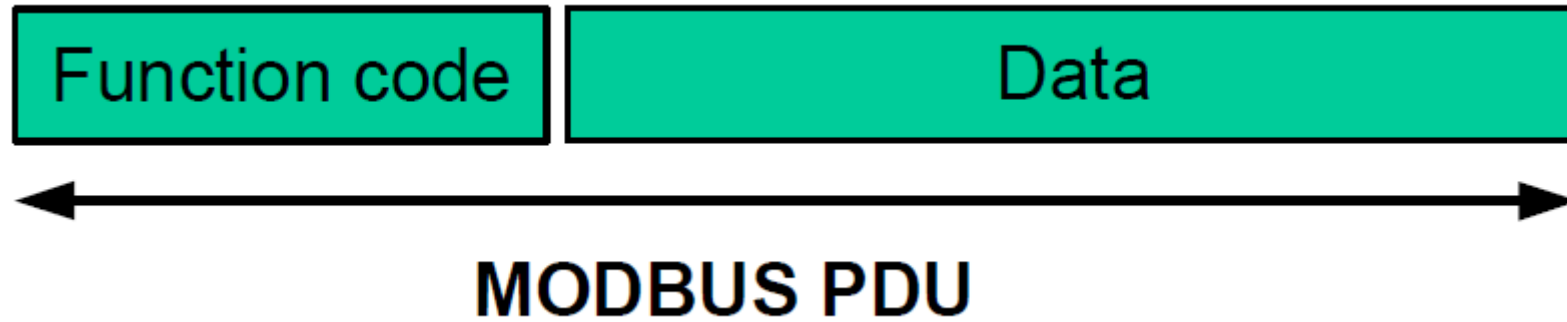
- 寄存器輸入

- 類比輸入點 (Analog Input)
 - 每位址 16 bit 寬

- 保持寄存器

- 類比輸出點 (Analog Output)
 - 每位址 16 bit 寬

Modbus 請求基本訊框



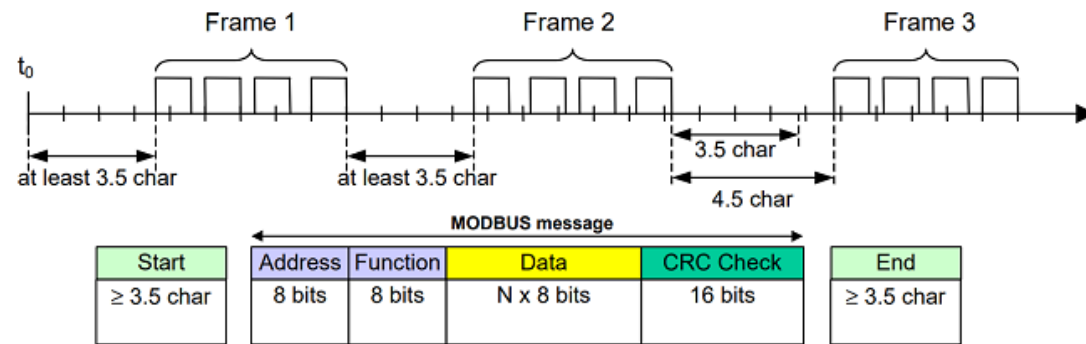
- Modbus 請求的最基本構成
 - 功能碼。
 - 資料。

Modbus 功能碼對應

				Function Codes	
				code	Sub code
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02	
		Internal Bits Or Physical coils	Read Coils	01	
			Write Single Coil	05	
			Write Multiple Coils	15	
	16 bits access	Physical Input Registers	Read Input Register	04	
			Read Holding Registers	03	
		Internal Registers Or Physical Output Registers	Write Single Register	06	
			Write Multiple Registers	16	
			Read/Write Multiple Registers	23	
			Mask Write Register	22	
			Read FIFO queue	24	
	File record access		Read File record	20	
			Write File record	21	
Diagnostics			Read Exception status	07	
			Diagnostic	08	00-18,20
			Get Com event counter	11	
			Get Com Event Log	12	
			Report Server ID	17	
			Read device Identification	43	14
Other			Encapsulated Interface Transport	43	13,14
			CANopen General Reference	43	13

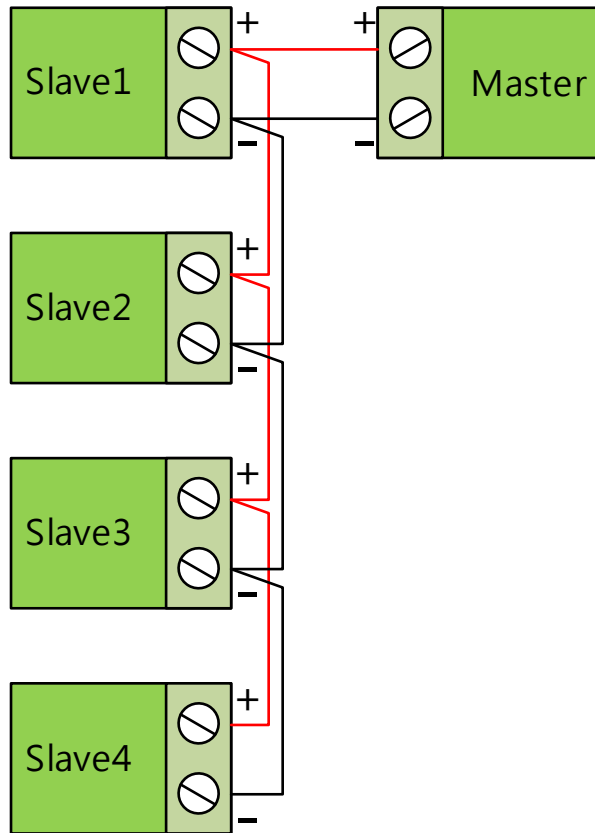
- 資料存取類功能碼對應於特定記憶體分區之行為，以下常用：
 - 01 讀取線圈
 - 02 讀取離散輸入
 - 03 讀取保持寄存器
 - 04 讀取寄存器輸入
 - 05 寫入單一線圈
 - 06 寫入單一寄存器
 - 15 寫入多個線圈
 - 16 寫入多個寄存器

Modbus RTU 訊框構成



- 每訊框（frame，幀）由 4 個部分構成
 - 從站位址（Address、Slave ID）
 - 功能代碼（Function Code）
 - 資料（Data）
 - 檢查碼（CRC）
- 其中從站位址與檢查碼是因應網路層尋找設備與驗證命令可靠性而擴展的部分。

Modbus RTU 常用實體層 - RS485 總線



- 兩線菊花鍊串聯。
- 以兩線電壓差分訊號通訊。
- 單主多從式半雙工廣播通訊。

Modbus RTU 通訊前設定

- 鮑率 (Baud Rate) : 即傳輸速度，9600、19200 bps 等
- 資料長度 (Data Bits) : 5 ~ 8 bits，最常見為 8 bits
- 同位校驗 (Parity) : 無校驗 (N) / 奇校驗 (O) / 偶校驗 (E)
- 停止位元 (Stop Bit) : 1、1.5、2 bits

Modbus RTU 讀取保持寄存器

01 03 40 00 00 01 91 CA

ADR FUN DATA

CRC

- 傳至 01 號設備
- 請求執行 03 號功能
 - 讀取保持寄存器 (Read Holding Registers)
- 內文為 40 00 00 01
 - 以 03 號命令的解釋法為，從 40 00 讀取 00 01 個寄存器。
- 檢查碼 91 CA

Modbus RTU 讀取保持寄存器之回傳

01 03 02 00 00 B8 44

ADR FUN DATA CRC

- 來自 01 號設備
- 回應 03 號功能
 - 讀取保持寄存器 (Read Holding Registers)
- 內文為 02 00 00
 - 以 03 號命令的解釋法為，共讀取 02 Byte 長，讀取內容為 00 00。
- 檢查碼 B8 44

Modbus RTU 寫入單一保持寄存器

01 06 40 00 00 01 5D CA

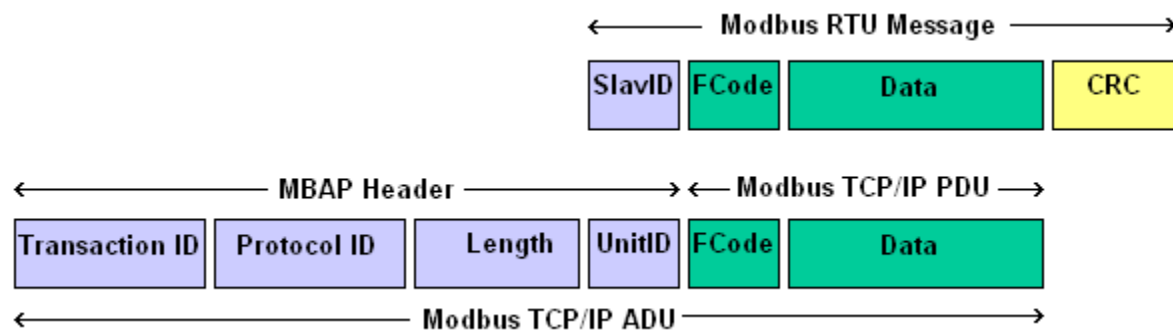
ADR FUN DATA

CRC

- 傳至 01 號設備
- 請求執行 06 號功能
 - 寫入保持寄存器 (Read Holding Registers)
- 內文為 40 00 00 01
 - 以 06 號命令的解釋法為，於 40 00 寫入 00 01 之內容。
- 檢查碼 5D CA

Modbus TCP

- Modbus 於 TCP/IP 協定下之擴展
- 慣例 TCP 通訊埠號為 502



Modbus TCP 網路封包

The image shows a Wireshark packet capture window titled '*Npcap Loopback Adapter'. The packet list pane shows two packets, 44 and 46, both of type Modbus/TCP. Packet 44 is selected, and its details pane shows the following structure:

- Frame 44: 108 bytes on wire (864 bits), 56 bytes captured (448 bits) on interface 0
- Null/Loopback
- Internet Protocol Version 4, Src: 192.168.127.1, Dst: 192.168.127.1
- Transmission Control Protocol, Src Port: 3973, Dst Port: 502, Seq: 1, Ack: 1, Len: 12
- Modbus/TCP
 - Transaction Identifier: 0
 - Protocol Identifier: 0
 - Length: 6
 - Unit Identifier: 1
- Modbus
 - .000 0011 = Function Code: Read Holding Registers (3)
 - Reference Number: 0
 - Word Count: 1

The packet bytes pane shows the raw data in hexadecimal and ASCII. The first 12 bytes are highlighted with red boxes, indicating the Modbus/TCP header and the function code (03) and unit identifier (01).

No.	Time	Source	Destination	Protocol	Length	Info
44	1.682211	192.168.127.1	192.168.127.1	Modbus/TCP	108	Query: Trans: 0; Unit: 1,
46	1.682373	192.168.127.1	192.168.127.1	Modbus/TCP	106	Response: Trans: 0; Unit: 1,

Frame 44: 108 bytes on wire (864 bits), 56 bytes captured (448 bits) on interface 0

Null/Loopback

Internet Protocol Version 4, Src: 192.168.127.1, Dst: 192.168.127.1

Transmission Control Protocol, Src Port: 3973, Dst Port: 502, Seq: 1, Ack: 1, Len: 12

Modbus/TCP

- Transaction Identifier: 0
- Protocol Identifier: 0
- Length: 6
- Unit Identifier: 1

Modbus

- .000 0011 = Function Code: Read Holding Registers (3)
- Reference Number: 0
- Word Count: 1

0000 02 00 00 00 45 00 00 34 08 cc 40 00 80 06 00 00E..4..@.....

0010 c0 a8 7f 01 c0 a8 7f 01 0f 85 01 f6 fa 8c 9c e6f..P..z.....

0020 ba cb 09 72 50 18 00 20 7f 7a 00 00 00 00 00 00rP.....

0030 00 06 01 03 00 00 00 01P.....

Modbus/TCP (mbtcp), 12 bytes

Packets: 53 · Displayed: 2 (3.8%) · Dropped: 0 (0.0%) Profile: Default

PyModbus

- 以 Python 實現的 Modbus 通訊套件。
- 支援 Modbus RTU、Modbus ASCII、Modbus TCP 協議。

Modbus RTU 讀值示例

```
from pymodbus.client.sync import ModbusSerialClient

# 建立 RTU 客戶端，以 9600 N 8 1 開啟樹莓派 /dev/ttyUSB0 通訊埠
client = ModbusSerialClient(method='rtu', port='/dev/ttyUSB0', baudrate=9600,
                             parity='N', bytesize=8, stopbits=1)

# 讀取站號 1 號自 0x4000 開始共 1 個保持寄存器，儲存於變數 response
response = client.read_holding_registers(address=0x4000, count=1, unit=0x01)

# 於終端機顯示讀值
print(response.registers[0])

# 關閉客戶端
client.close()
```

Modbus RTU 寫值示例

```
from pymodbus.client.sync import ModbusSerialClient

# 建立 RTU 客戶端，以 9600 N 8 1 開啟樹莓派 /dev/ttyUSB0 通訊埠
client = ModbusSerialClient(method='rtu', port='/dev/ttyUSB0', baudrate=9600,
                             parity='N', bytesize=8, stopbits=1)

# 寫入站號 1 號 0x4000 之保持寄存器，內容為 0x0001
response = client.write_register(address=0x4000, value=0x0001, unit=0x01)

# 關閉客戶端
client.close()
```

Modbus TCP 讀值示例

```
from pymodbus.client.sync import ModbusTcpClient

# 建立 TCP 客戶端，開啟 192.168.0.254:502 通訊埠
client = ModbusTcpClient(host='192.168.0.254', port=502)

# 讀取站號 1 號自 0x4000 開始共 1 個保持寄存器，儲存於變數 response
response = client.read_holding_registers(address=0x4000, count=1, unit=0x01)

# 於終端機顯示讀值
print(response.registers[0])

# 關閉客戶端
client.close()
```

Modbus TCP 寫值示例

```
from pymodbus.client.sync import ModbusTcpClient

# 建立 TCP 客戶端，開啟 192.168.0.254:502 通訊埠
client = ModbusTcpClient(host='192.168.0.254', port=502)

# 寫入站號 1 號 0x4000 之保持寄存器，內容為 0x0001
response = client.write_register(address=0x4000, value=0x0001, unit=0x01)

# 關閉客戶端
client.close()
```

PyModbus 自學

- <https://pymodbus.readthedocs.io/en/latest/> 官網
 - 有完整的文件記載套件內所有類型與方法。
 - 提供大量的範例代碼，包含通步非同步客戶端、遠端轉發器等。